International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.460.1

(03/2013)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Supplementary services for multimedia

Guidelines for the use of the generic extensible framework

Recommendation ITU-T H.460.1

ITU-T

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Supplementary services for multimedia	**H.450–H.499**
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
Ubiquitous sensor network applications and Internet of Things	H.640–H.649
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	H.760–H.769
IPTV service discovery up to consumption	H.770–H.779
Digital Signage	H.780–H.789

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.460.1

Guidelines for the use of the generic extensible framework

Summary

Recommendation ITU-T H.460.1 gives guidelines on how to use the "Generic Extensible Framework" specified in Recommendation ITU-T H.323. It describes when to use the framework, and how to specify features using the framework. It also gives examples on how the framework's negotiation scheme works.

This revision introduces a number of corrections and clarifications by incorporating technical and editorial corrections from the ITU-T H.323-series Implementers Guide (03/2011).

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T H.460.1	2002-03-29	16
2.0	ITU-T H.460.1	2013-03-16	16

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T H.460.1

Guidelines for the use of the generic extensible framework

1 Scope

This Recommendation provides information on when and how to use the generic extensible framework (GEF) defined in [ITU-T H.323]. It does not repeat the text from [ITU-T H.323]. Instead, it expands and elaborates on the text in [ITU-T H.323] and thus should be read in conjunction with [ITU-T H.323] rather than being viewed as a substitute to it.

The main topics covered by this Recommendation are: when to use GEF, how to specify GEF modules, and examples of the GEF negotiation mechanism.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T H.225.0] Recommendation ITU-T H.225.0 (2009), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

[ITU-T H.323] Recommendation ITU-T H.323 (2009), *Packet-based multimedia communications systems*.

3 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ABNF	Augmented Backus-Naur Form
ASN.1	Abstract Syntax Notation One
GEF	Generic Extensible Framework
XML	Extensible Markup Language

4 What is GEF

The generic extensible framework (GEF) provides a low overhead way of adding functionality to [ITU-T H.323] without adding to the ASN.1 base specification of [ITU-T H.225.0]. GEF provides a common feature negotiation mechanism, and the ability to carry opaque data in all ITU-T H.225.0 and Annex G messages of [ITU-T H.225.0]. Thus it allows application specific extensions to be made to the ITU-T H.323 suite of Recommendations without burdening all ITU-T H.323 implementations with all specified extensions.

GEF is explained formally in the main body of [ITU-T H.323].

5 When to use GEF

New functionality can be added to [ITU-T H.323] by adding new syntax to the ITU-T H.225.0 ASN.1 base definition, defining a GEF module, or defining an ITU-T H.450 supplementary service. When deciding which method to use, it may be appropriate to consider the feature negotiation aspects and the data transport aspects of the feature separately. Thus, in some cases, it may be appropriate to implement a feature primarily as an ITU-T H.450 supplementary service, but negotiate the feature using GEF's feature negotiation capabilities.

To decide which method should be used to add new functionality, the following guidelines are suggested.

- If the feature is applicable to a large number of applications, then it may be appropriate to add it to the ITU-T H.225.0 ASN.1 base specification directly. Note that, as [ITU-T H.323] is now fairly mature, it is not expected that such features will be encountered and a strong case will need to be made for taking this course of action. The most likely situation in which this will occur is in the case of extending already existing parameters such as AliasAddress and TransportAddress.
- Or if the feature only requires a few parameters, then define it as a GEF module.
- Or if the feature requires data to be carried in ITU-T H.225.0 RAS or Annex G of [ITU-T H.225.0], then define it as a GEF module.
- Or if the feature contains numerous parameters, then define it as either ASN.1 (or similar method such as ABNF or XML) to be encoded into a raw element of a GEF **content** construct, or define an ITU-T H.450 supplementary service.

These considerations are captured in the flow chart shown in Figure 1 below.

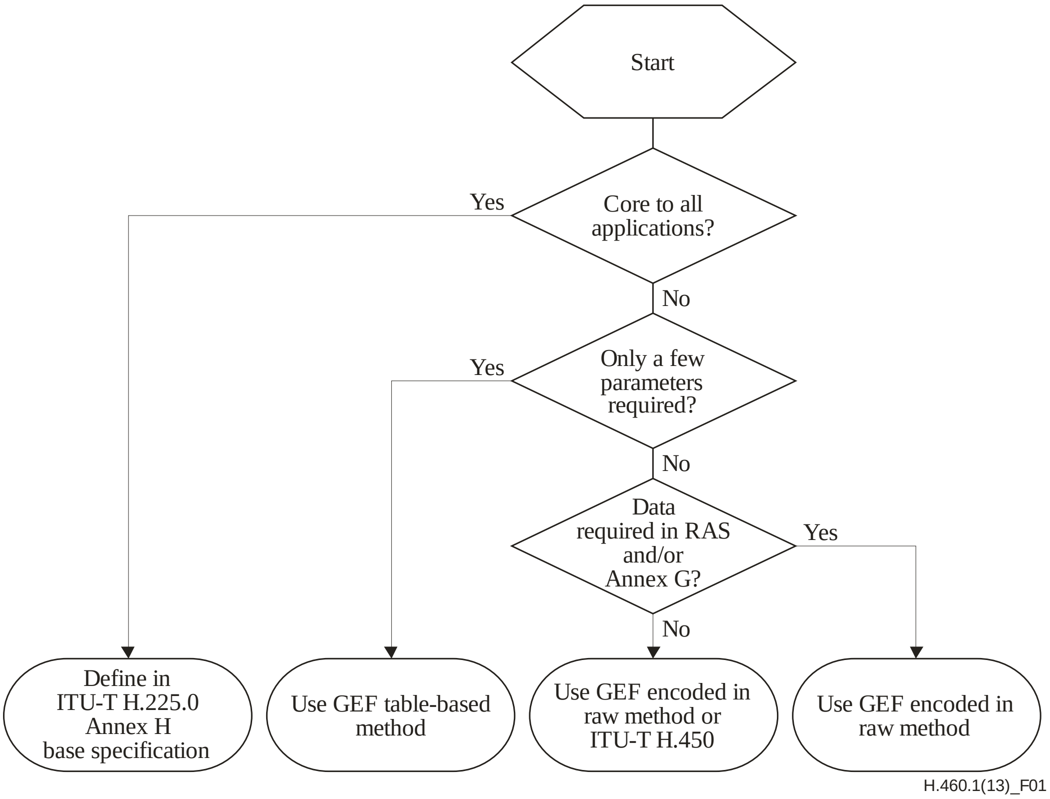

Start
Yes
Core to all applications?
No
Yes
Only a few parameters required?
No
Data required in RAS and/or Annex G?
Yes
No
Define in ITU-T H.225.0 Annex H base specification
Use GEF table-based method
Use GEF encoded in raw method or ITU-T H.450
Use GEF encoded in raw method
H.460.1(13)_F01

Figure 1 – Decision tree for how to specify a new feature

6 GEF module identification

GEF allows for both ITU-T approved and non-ITU-T approved module definitions. All ITU-T approved GEF module definitions shall be defined as part of the ITU-T H.460 series of Recommendations and shall use the **standard** variant of **id** in **GenericData** to identify them. All non-ITU-T approved modules shall use either the **oid** or **nonStandard** variants of **id** in **GenericData** to identify them.

There is a direct relationship between an ITU-T H.460 series Recommendation's identity and the value encoded into the **standard** variant of **id** in **GenericData**. That is Recommendation ITU-T H.460.n (where n is an integer number) shall use the value n in the **standard** variant of **id** in **GenericData** to identify the module. For example, ITU-T H.460.2 shall have the **standard** variant of **id** in **GenericData** set to 2.

7 Specifying GEF modules

GEF modules may be specified in a number of ways, clarity of definition being the principle concern. To provide some consistency of specification, this clause describes two ways in which GEF modules can be defined, these being the table-based method and the encoded in raw method.

Note that if a feature requires both negotiation and opaque data transfer, it may be necessary to document the requirements for each in separate sections of the specification.

7.1 Table-based method

In the table-based method, each of the parameters constituting a feature is described in tabular form. Each parameter definition maps to a separate **EnumeratedParameter** for data exchange. Note that the tables are not expected to capture all the detail of the feature's use and operation and additional explanatory text would typically be expected.

An instance of Table 1 is used to describe the feature as a whole. This captures the features name, its identifier and a brief description. A single instance of this table suffices for features that contain both negotiation and data transport.

Table 1 – Tabular specification of a feature

Feature name	**The name of the feature**
Feature Description:	Short description of the feature. This may be augmented by text elsewhere in the document describing the feature.
Feature identifier type:	Indicates whether the identifier is Standard, oid, or nonStandard.
Feature identifier value:	The actual value of the identifier.

Each parameter used by the feature is described using an instance of Table 2. Once again, this includes the parameters name, identifier and description as well as other information. Parameters conveyed within compound or nested parameters should be described in a separate section of the specification. If parameters are needed for both the negotiation and data transport aspects of the feature, it may be appropriate to have the two sets of parameters captured in different sections of the specification describing the feature.

Table 2 – Tabular specification of a parameter

Parameter name	The name of the parameter
Parameter description:	Short description of the parameter
Parameter identifier type:	Indicates whether the identifier is Standard, oid, or nonStandard
Parameter identifier value:	The actual value of the identifier
Parameter type:	One of raw, text, unicode, bool, number8, number16, number32, id, alias, transport, compound, nested.
Parameter cardinality:	How many times the parameter may occur

7.1.1 Example definition of a fictitious feature using the table-based method

This clause gives a fictitious example of a feature using the table-based method of feature definition.

Table 3 – Specification of the fictitious example feature

Feature name	Fictitious example
Feature description:	This is a fictitious example to illustrate how to use the table-based method to specify a GEF module.
Feature identifier type:	Standard
Feature identifier value:	1

Table 4 – Parameters for the fictitious example feature

Parameter name	priority
Parameter description:	Describes the priority of this call in relation to other calls that may be received.
Parameter identifier type:	Standard
Parameter identifier value:	0
Parameter type:	number8
Parameter cardinality:	Once and once only

Parameter name	destination
Parameter description:	Identity of the remote party that is being contacted.
Parameter identifier type:	Standard
Parameter identifier value:	1
Parameter type:	alias
Parameter cardinality:	One or more

Parameter name	message-display-lines
Parameter description:	One of more message lines that should be displayed when the call is being processed.
Parameter identifier type:	Standard
Parameter identifier value:	2
Parameter type:	text
Parameter cardinality:	Zero or more

7.2 Encoded in raw method

The second method of GEF module definition defines a feature using existing message encoding methods, such as ASN.1, XML or ABNF. In this case, message fragments are encoded according to the normal rules of the chosen specification method and exchanged as the raw variant of a GEF parameter. This is particularly recommended for features that contain a sufficient number of parameters to make the table-based method of definition cumbersome.

In addition to specifying the content of the GEF parameter, it is also necessary to specify the identifier of the GEF parameter. This can be done using the table-based method of specification, or in narrative text.

If the feature is defined using ASN.1, then it is recommended that the basic aligned variant of the PER encoding rules be used. However, irrespective of this, encoding rules that are used, if different from the above, shall be explicitly stated in the specification of the feature.

Defining features using ABNF potentially allows features to inter-operate unmodified with other signalling protocols that perform a similar function to [ITU-T H.323] (e.g., SIP). Note that such a representation will still have to be conveyed in a raw element to allow for UTF-8 encoding of characters.

Appendix I

Sample GEF message exchanges

(This appendix does not form a normative part of this Recommendation.)

I.1 Examples of GEF negotiation

This appendix illustrates various aspects of GEF's negotiation mechanism. It is for illustrative purposes only.

I.1.1 RAS scenarios

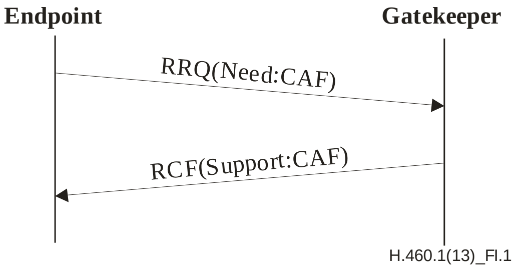
The figures shown below illustrate an endpoint negotiating with a gatekeeper.

For illustration purposes, the negotiation is shown using the RRQ/RCF/RRJ message set. However, GEF is not limited to this set of messages for negotiation.

The example assumes that the endpoint wants to use a fictional charge advice feature, called CAF. The details of this are not important, but the principle would be that the gatekeeper would supply the endpoint with some notification of the call charge at the end of a call using the DCF message.

Similarly, the gatekeeper wishes to use a bandwidth management feature called BWF. Again, the details of this feature are not important, but the principle would be that the parameters within the module would define some form of bandwidth template that the endpoint must adhere to.

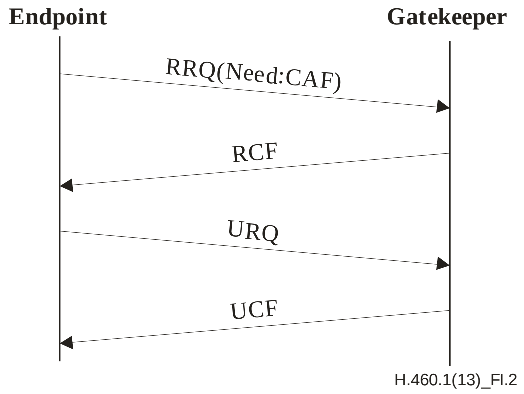
The following diagrams illustrate various aspects of the negotiation process.



Conditions:
– **Endpoint** needs/requires CAF feature.
– **Gatekeeper** supports CAF feature.
Request successful.

Figure I.1 – Endpoint signalling to indicate the needed CAF feature

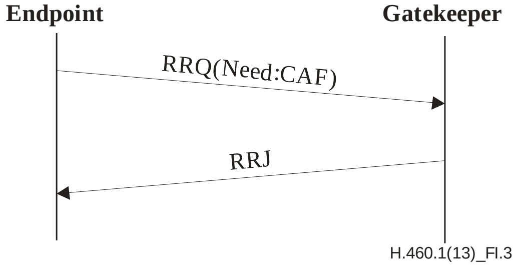
Description: In Figure I.1 the endpoint requests the CAF feature, and the gatekeeper indicates that it supports the CAF feature. The negotiation has been successful and the endpoint may now use the feature.



Conditions:
– **Endpoint** needs/requires CAF feature.
– **Gatekeeper** does not support the framework.
Procedure fails.
(**Endpoint** has to handle backwards compatibility when it gets the reply)

Figure I.2 – Endpoint unregisters if the gatekeeper lacks the needed feature

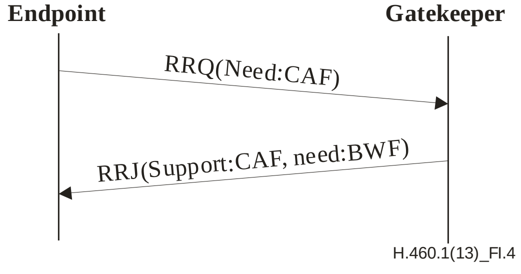
Description: In Figure I.2 the endpoint requests the CAF feature, but the gatekeeper does not support the GEF negotiation mechanism and so does not see the request for the CAF feature. As the gatekeeper does not see the request for the CAF feature, it believes that it can accept the request and sends an RCF. The endpoint observes that the gatekeeper has not signalled support for CAF and therefore decides to unregister as the gatekeeper does not provide the features it requires.



Conditions:
– **Endpoint** needs CAF feature.
– **Gatekeeper** does not support CAF feature.
Request fails.

Figure I.3 – Gatekeeper rejects the RRQ if it does not support the needed CAF feature

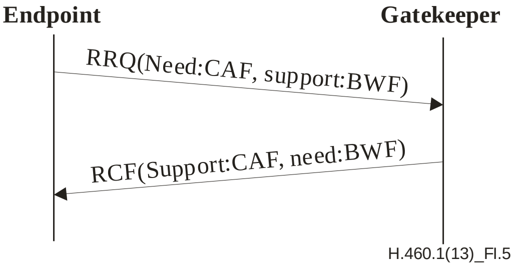
Description: Figure I.3 is similar to the Figure I.2 case, except that the gatekeeper supports the GEF negotiation mechanism. The gatekeeper knows that it does not support the CAF feature and so rejects the request with an RRJ.



Conditions:
– **Endpoint** needs CAF feature.
– **Gatekeeper** supports CAF, BUT also needs BWF feature.
Request fails.

Figure I.4 – Gatekeeper accepts the RRQ if it supports the needed CAF feature

Description: In the case shown in Figure I.4, the endpoint needs the CAF feature and the gatekeeper needs the BWF feature. The endpoint indicates that it needs the CAF feature in the RRQ. The gatekeeper sees that CAF is needed, which it supports, but it does not see that the endpoint supports the BWF feature. The gatekeeper therefore rejects the request, indicating that it supports the CAF feature, but needs the BWF feature.



Conditions:
– **Endpoint** needs CAF and supports BWF.
– **Gatekeeper** supports CAF and needs BWF.
Request successful.

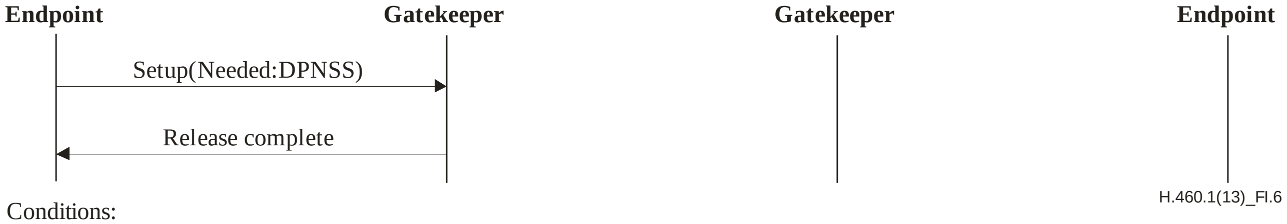
Figure I.5 – Gatekeeper indicates support for the needed CAF feature and indicated the need for the endpoint to support the BWF feature

Description: In Figure I.5, the endpoint signals that it needs the CAF feature, and also supports the BWF feature. As the gatekeeper supports the CAF feature, it signals this in its reply. The gatekeeper also needs the endpoint to use the BWF feature, and signals this as needed in its reply.

I.1.2 Call signalling scenarios

The following figures show examples of GEF call signalling negotiation. The gatekeeper routed model is shown on the basis that the direct model is a special (simplified) case of the gatekeeper routed model.

In these examples the fictitious features *DPNSS*, *450Proxy* and *Account* are used. The details of what these features entail is not important to the discussion here and are therefore not described.



Conditions:
– **Initiating endpoint** needs DPNSS feature.
– **First gatekeeper** does not support feature.
Unsuccessful.

Figure I.6 – Gatekeeper rejects a SETUP message that requires an unsupported feature

Description: Figure I.6 shows the endpoint initiating a call that requires the fictional DPNSS feature. The first gatekeeper supports GEF negotiation, but does not support the DPNSS feature and therefore rejects the call.

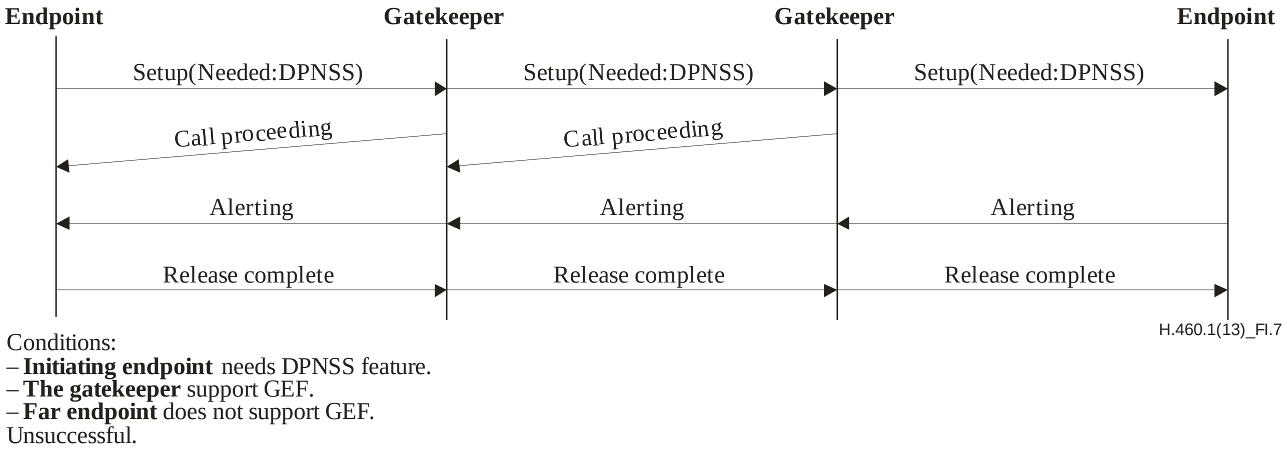


Figure I.7 – The calling endpoint terminates a call when it learns the called endpoint does not indicate support for the needed DPNSS feature

Description: Figure I.7 shows the endpoint initiating a call that requires the DPNSS feature. While the gatekeepers support GEF, the far endpoint does not. It is therefore ignorant of the need for the DPNSS feature and therefore progresses the call by sending an ALERTING message. The ALERTING message does not contain an indication for support of the DPNSS feature and therefore the initiating endpoint terminates the call.

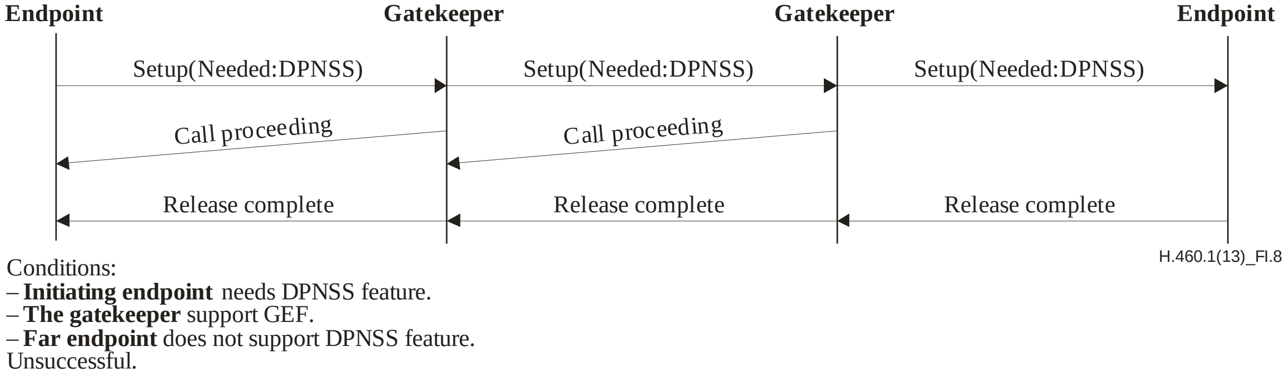


Figure I.8 – Called endpoint rejects a call that needs the DPNSS feature since it is not supported

Description: In Figure I.8 the endpoint wishes to create a call using the DPNSS feature. The remote endpoint supports GEF negotiation, but does not support the DPNSS feature and so rejects the call.

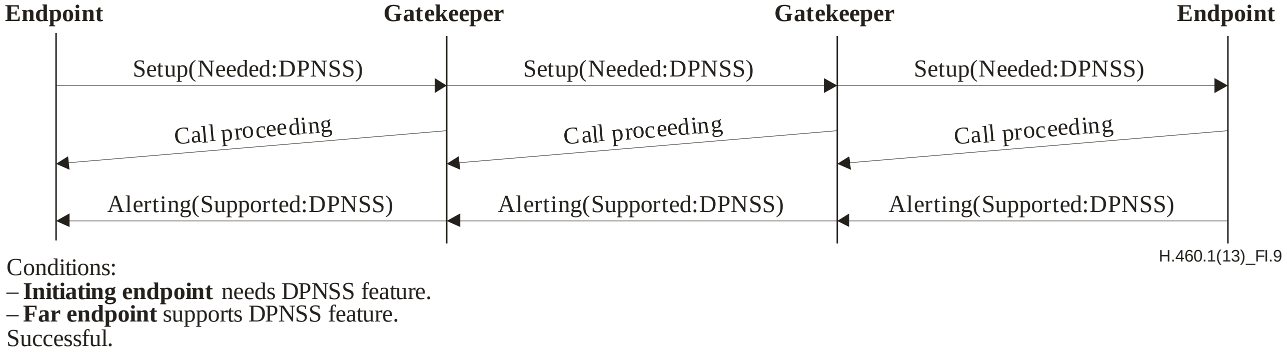
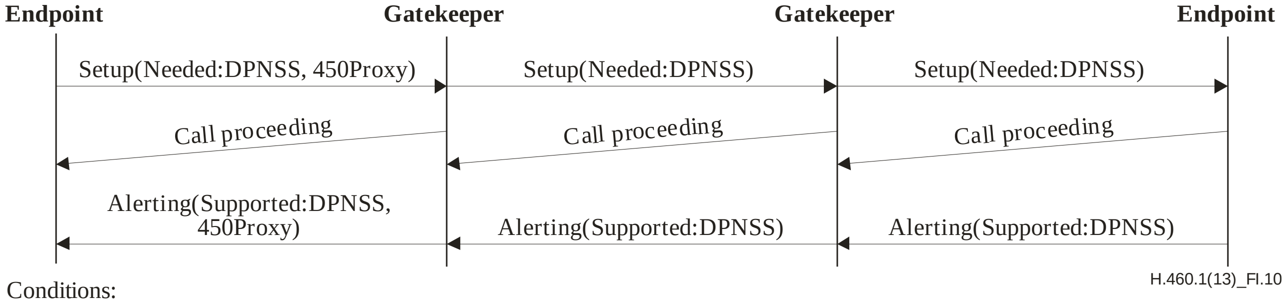


Figure I.9 – The called endpoint indicates support for the needed DPNSS feature

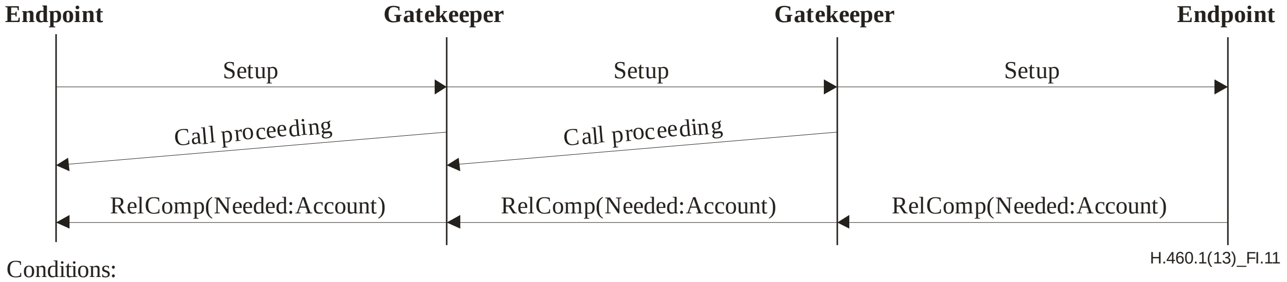
Description: In the case shown in Figure I.9, the remote endpoint does support the DPNSS feature. It is therefore able to accept the call. It signals support for DPNSS in ALERTING, which is the first end-to-end message.



Conditions:
– **Initiating endpoint** needs DPNSS and 450Proxy feature.
– **Local gatekeeper** supports 450Proxy feature.
– **Far endpoint** supports DPNSS feature.
Successful.

Figure I.10 – The first gatekeeper removes a needed 450Proxy feature requested by the calling endpoint that is handled by the gatekeeper

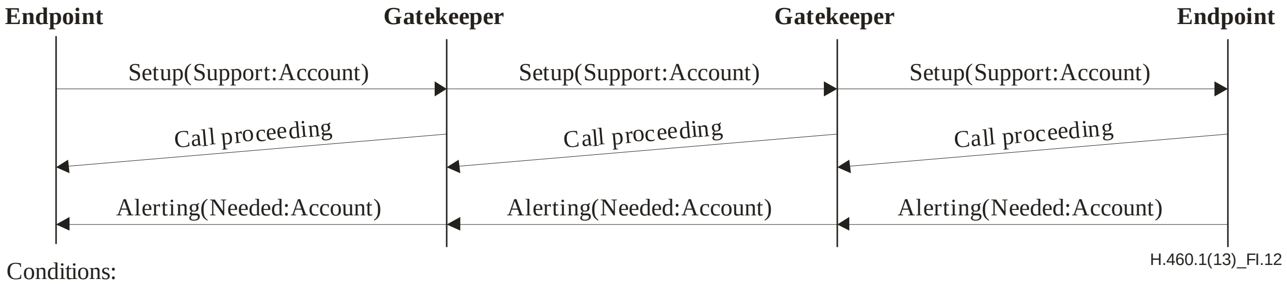
Description: Figure I.10 is similar to the case shown in Figure I.9, except that the endpoint also wishes the ITU-T H.450 proxing feature (identified as 450Proxy) to be supported. The first gatekeeper accepts responsibility for this, and removes the needed feature from the message before routing it onwards. The far endpoint implements the DPNSS feature and sends back an ALERTING message indicating this. When the ALERTING message passes through the first gatekeeper, it modifies the message indicating its ability to implement the ITU-T H.450 proxy feature.



Conditions:
– **Initiating endpoint** does not support framework, or has no features.
– **Far endpoint** needs account feature.
Unsuccessful.

Figure I.11 – The called endpoint rejects a call, indicating it needs an account feature

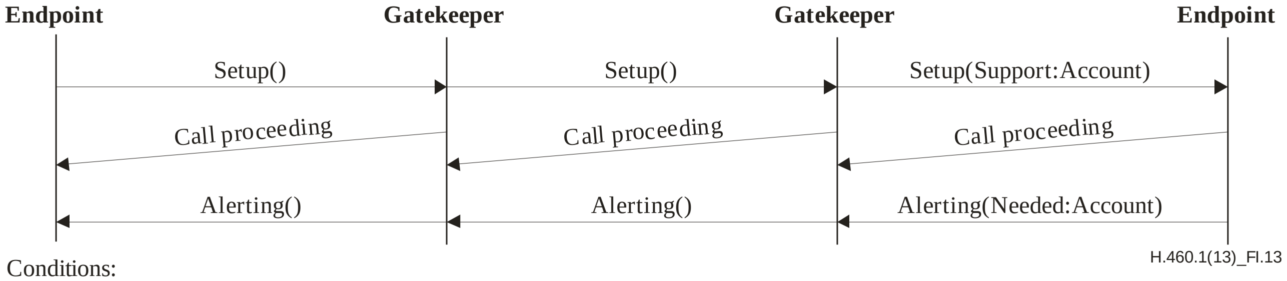
Description: In Figure I.11 the endpoint initiates a call to an endpoint that requires an account feature to be specified. As the calling endpoint did not indicate that it supports the account feature, the far endpoint clears the call.



Conditions:
– **Initiating endpoint** supports account feature.
– **Far endpoint** needs account feature.
Successful.

Figure I.12 – The calling endpoint indicates support for the account feature and the called endpoint indicates that the feature is needed

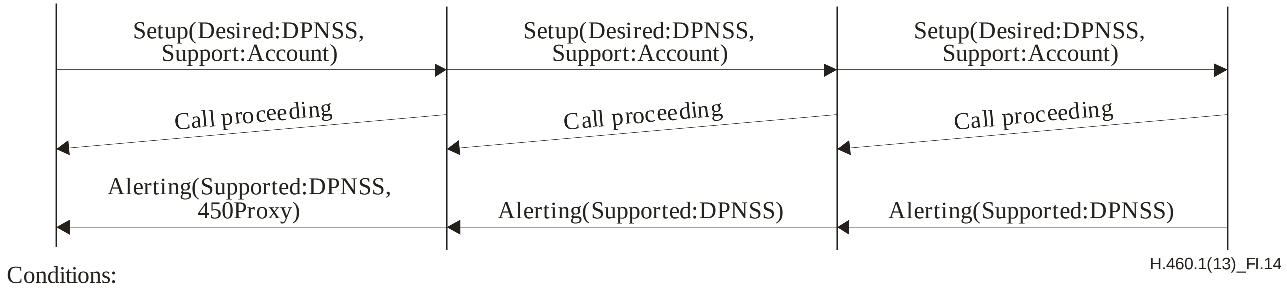
Description: Figure I.12 shows a case in which the calling endpoint indicates that it supports the account feature. The called endpoint needs this feature and as support for the feature is indicated in the setup message, it is able to accept the call.



Conditions:
– **Initiating endpoint** does not support account feature.
– **Second gatekeeper** supports account feature.
– **Far endpoint** needs account feature.
Successful.

Figure I.13 – The second gatekeeper inserts support for the account feature and the called endpoint indicates a need for that feature

Description: Figure I.13 shows a gatekeeper adding its own supported feature set. It is similar to the previous example. Again the called endpoint needs the account feature, but the account feature is not signalled by the originating endpoint. In this case the last gatekeeper knows that it is necessary to support the account feature for the call to be accepted by the terminating endpoint, so it indicates that it supports this feature. As the account feature is supported, the called endpoint is able to accept the call.

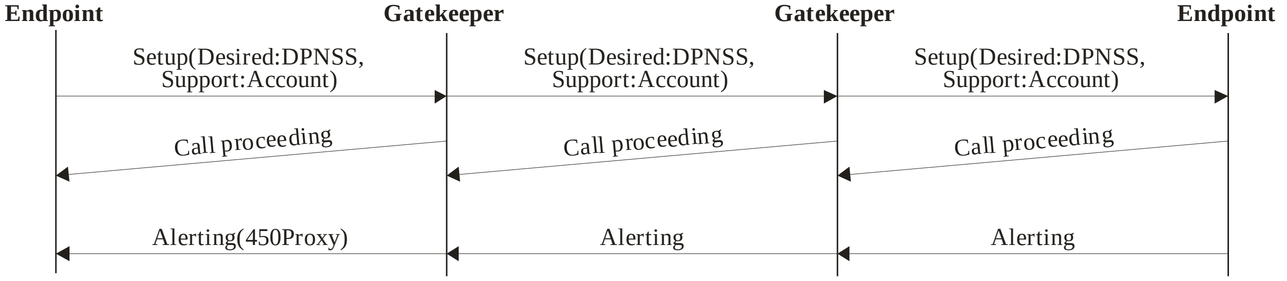

Endpoint
Gatekeeper
Gatekeeper
Endpoint
Setup(Desired:DPNSS, Support:Account)
Setup(Desired:DPNSS, Support:Account)
Setup(Desired:DPNSS, Support:Account)
Call proceeding
Call proceeding
Call proceeding
Alerting(Supported:DPNSS, 450Proxy)
Alerting(Supported:DPNSS)
Alerting(Supported:DPNSS)
H.460.1(13)_FI.14

Conditions:
– **Initiating endpoint** would like DPNSS feature, but does not need it.
The initiating endpoint also supports the account feature.
– **Far endpoint** supports the DPNSS feature.
– **First gatekeeper** supports 450Proxy feature and declares it, even though it is not asked for.
Successful.

Figure I.14 – Calling endpoint indicates a desire for the DPNSS feature

Description: In Figure I.14, the calling endpoint desires the DPNSS feature, but does not need it. The called endpoint supports the DPNSS feature, and therefore signals this in its ALERTING reply.

NOTE 1 – In this case the first gatekeeper adds in signalling support for the 450Proxy feature even though it was not asked for. This is legitimate and is typically the easiest way to implement indicating supported features.

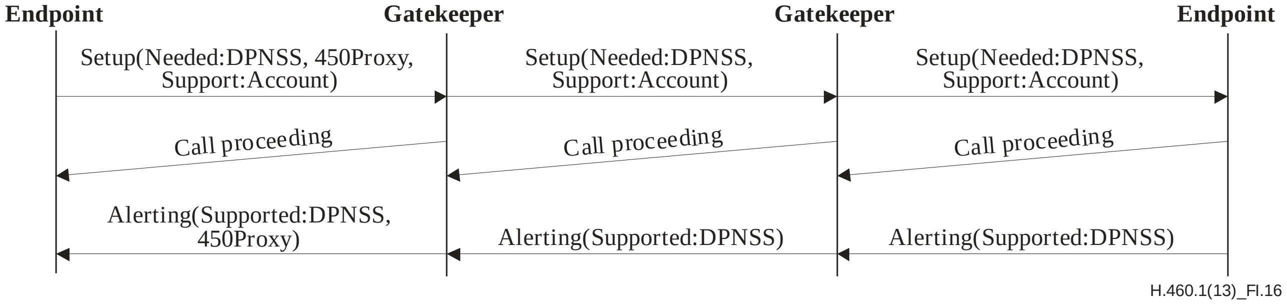


Conditions:
– **Initiating endpoint** would like DPNSS feature, but does not need it.
The initiating endpoint also supports the account feature.
– **Far endpoint** does NOT support the DPNSS feature.
– **First gatekeeper** supports 450Proxy feature and declares it, even though it is not asked for.
Successful.

Figure I.15 – The calling endpoint indicates a desired and supported feature, neither of which is supported by the called endpoint

Description: The case shown in Figure I.15 is a similar scenario to that shown in Figure I.14, except that in this case the called endpoint does not support the DPNSS feature. However, the call can still proceed as it is only a desired feature. The calling endpoint knows that the DPNSS feature is not supported because it is not indicated in the ALERTING message that comes back.

NOTE 2 – In this case the first gatekeeper adds in signalling support for the 450Proxy feature even though it was not asked for. This is legitimate and is typically the easiest way to implement indicating supported features.



Endpoint
Gatekeeper
Gatekeeper
Endpoint
Setup(Needed:DPNSS, 450Proxy, Support:Account)
Setup(Needed:DPNSS, Support:Account)
Setup(Needed:DPNSS, Support:Account)
Call proceeding
Call proceeding
Call proceeding
Alerting(Supported:DPNSS, 450Proxy)
Alerting(Supported:DPNSS)
Alerting(Supported:DPNSS)
H.460.1(13)_FI.16

Figure I.16 – A combination of the negotiation elements

Description: Finally, Figure I.16 illustrates a number of aspects of the call signalling negotiation mechanism operating together. The originating endpoint needs the DPNSS feature in the remote endpoint and the 450Proxy feature in its local gatekeeper. It also signals support for the account feature. The local gatekeeper can implement the 450Proxy feature and so removes the 450Proxy needed feature from the message before passing it on. The remote endpoint observes that the DPNSS feature is needed and it supports it. It responds to the call by sending an ALERTING message, indicating that it supports the DPNSS feature. When the message gets to the originating endpoint's local gatekeeper, it signals that it supports the requested 450Proxy feature by adding this as a supported feature to the ALERTING message. The originating endpoint knows that the two features it needs are supported as it sees both of them advertised in the ALERTING message that it receives.

Bibliography

The following citations are for informative purposes only and do NOT constitute provisions of this Recommendation.

[b-ITU-T X.680] Recommendation ITU-T X.680 (2008) | ISO/IEC 8824-1:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

[b-ITU-T X.681] Recommendation ITU-T X.681 (2008) | ISO/IEC 8824-2:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.

[b-ITU-T X.691] Recommendation ITU-T X.691 (2008) | ISO/IEC 8825-2:2008, *Information technology – ASN.1 encoding rules – Specification of Packed Encoding Rules (PER)*.

[b-IETF RFC 2234] IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF*.

[b-IETF RFC 2279] IETF RFC 2279 (1998), *UTF-8, a transformation format of ISO 10646*.

[b-IETF RFC 2543] IETF RFC 2543 (1999), *SIP: Session Initiation Protocol*.

[b-W3C REC-xml] W3C REC-xml, October 2000, *Extensible Markup Language (XML) 1.0* (Second Edition).

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2013